

“The wallpaper is ugly”: Indoor Localization using Vision and Language

Seth Pate and Lawson L.S. Wong



Fig. 1. **Use of localization in robotics.** Localization is a necessary first step when a robot must help a human without perfect knowledge of their location. This may apply to search and rescue (top) or household assistance (bottom). In this paper, we study only the localization task. Photo credits: Ian Howard (top), Matterport3D (bottom).

Abstract—We study the task of locating a user in a mapped indoor environment using natural language queries and images from the environment. Building on recent pretrained vision-language models, we learn a similarity score between text descriptions and images of locations in the environment. This score allows us to identify locations that best match the language query, estimating the user’s location. Our approach is capable of localizing on environments, text, and images that were not seen during training. One model, finetuned CLIP, outperformed humans in our evaluation.

I. INTRODUCTION

Natural language is an important medium of communication between humans and robots [1]. Many robot tasks refer to, and rely on understanding, the robot’s spatial environment. Connecting natural language with the robot’s spatial knowledge is therefore critical. However, making this connection is challenging because humans tend to represent and express knowledge about the environment differently from robots. Natural language descriptions tend to be high-level, sparse, and semantic, whereas robot spatial representations are low-level, dense, and geometric.

In this paper, we propose and study one important problem in this intersection: locating a user in a known environment, given a natural language description of a desired location. This is an important capability in several robot applications, such as finding someone who is lost by asking them to describe their surroundings, or autonomous delivery, where

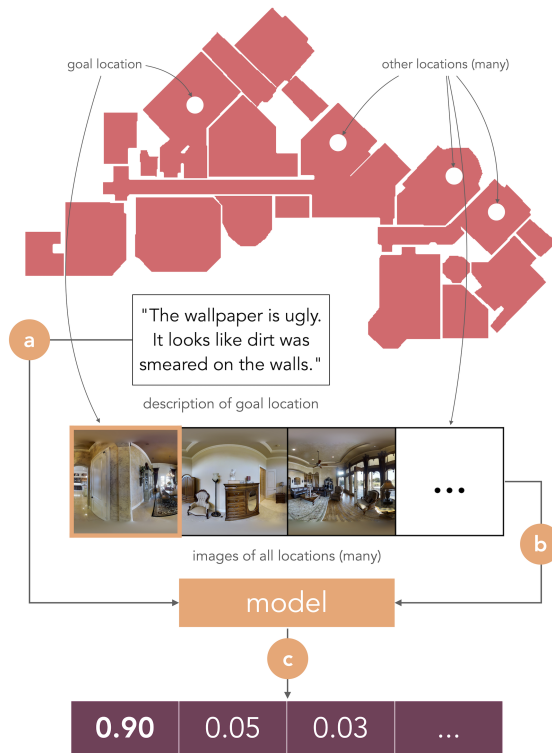


Fig. 2. **Vision-language localization.** (a) The model encodes the user’s description of their location, the goal. (b) The model encodes an exhaustive sample of images representing all locations in the environment. (c) The model produces a similarity score between each image and the description, which, after softmax, outputs a distribution to predict the user’s location.

the robot needs to communicate with and find the recipient via natural language. Location is also the first step in providing spatial directions to a user – before you can tell someone where to go, you need to know where they are.

We illustrate our task and high-level approach in Fig. 2. We assume that the robot has already mapped the environment and knows all the locations that may be queried. We use Matterport3D [2] as a source of mapped indoor environments with rich visual information. The user, located at some point in the environment, describes their surroundings in natural language. With this sentence and images from a discrete set of locations in the mapped environment, we compute a learned similarity score between the description query and each location’s images. We can use the resulting distribution of scores to estimate the user’s location.

To relate descriptions to images, we use a large pretrained vision-language model [3], CLIP [4], which learns complex

¹Khoury College of Computer Sciences, Northeastern University, Boston, MA pate.s@northeastern.edu ; lsw@ccs.neu.edu

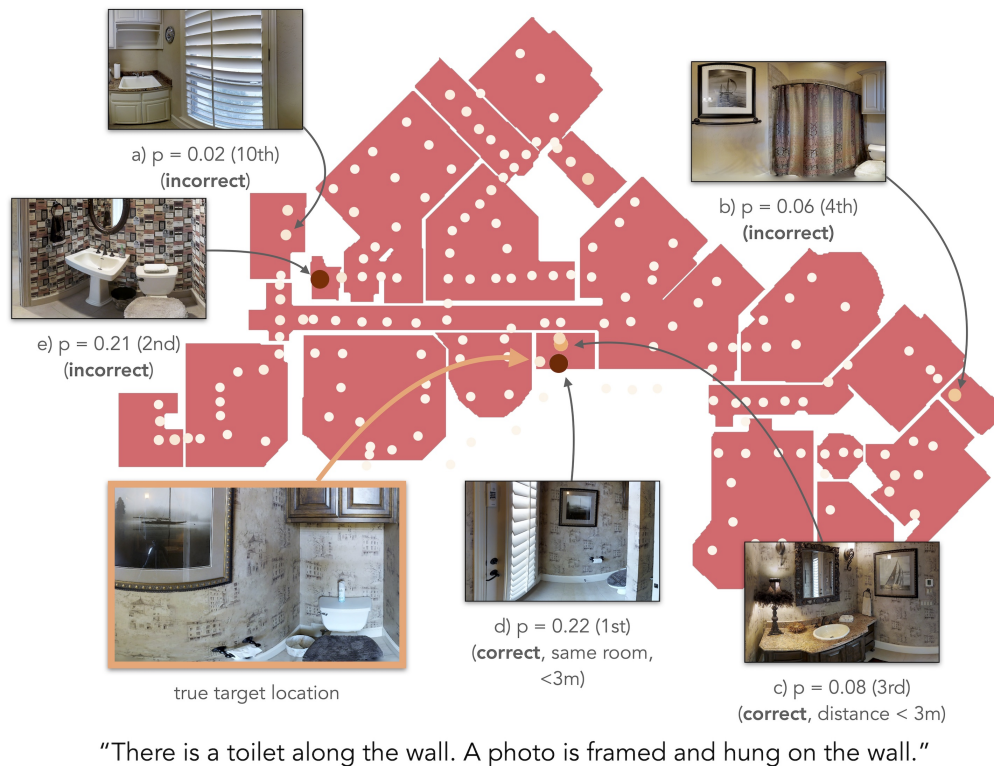


Fig. 3. **Example Model Output.** Our model creates a likelihood distribution across the 170 locations in this *scan*. The model’s confidence is shown by both the size and color of the circles, which represent *views*. We highlight some guesses alongside the true target location, a bathroom. From top left, clockwise: (a) The 10th guess is a laundry room with a sink, but no toilet. (b) The 4th guess has a photo and a toilet. It is a good guess, but the wrong bathroom. (c) The 3rd guess was taken from the hallway, but has a clear view of the target location. (d) The model’s best guess is in the same annotated region (room), adjacent to the target. (e) The 2nd guess is in another bathroom without a framed photo, only a mirror which may resemble one.

representations of images and text on different pretraining tasks where data is widely available.

To improve the model’s performance, we repurposed finetuning data from datasets that are related to our task. Specifically, we used data for vision and language navigation [5], [6], [7], [8] to construct two additional datasets for our task (‘RxR’ and ‘RxR_landmarks’), which gave significant improvements.

To evaluate the model, we gathered a small test dataset of human descriptions for locations in Matterport3D. Then we performed two experiments. First, we compared different finetuning subsets, as well as an alternate model using convolutional neural nets (CNNs) and a long short-term memory (LSTM) text encoder. Then we compared the best finetuned model to a human baseline on the test set. In this setting, the model outperformed the human baseline.

In summary, we define the task of vision-language localization, consider a simple approach using existing pre-trained vision-language models, collect and construct several datasets to finetune the pretrained model, and evaluate our models on localizing in Matterport3D indoor environments.

II. RELATED WORK

Natural language is widely used in robotics; see Tellex *et al.* [1] for a comprehensive survey. As the survey indicates, the primary focus of previous work has been on instruction

following and answering/asking questions. ‘Vision and language navigation’ [5] (VLN) extends the instruction following concept directly to sighted agents moving within indoor or outdoor environments [9], [10], [11], [12], [13]. A lot of VLN work, including ours, is done with the Matterport3D environment, which we describe below. [2]

Although many VLN papers ask embodied agents to navigate a certain path between two points, others simply ask agents to find an object or location within an environment, which is very similar to our location task. Examples include REVERIE [14] and SOON [15], whose benchmarks use deep recurrent neural networks, reinforcement learning, imitation learning, and graph embeddings to approach the problem.

Whereas these papers treat location as a navigation problem and ask agents to produce a series of actions, we simply ask agents to rank target locations in a known environment. This is more similar to Hahn *et al.* [16] and Chen *et al.* [17], who encode language instructions alongside a given environment to produce, respectively, a distribution of locations or a bounding box in that environment.

Our approach is different, in that we take environments as a set of unrelated images, and treat location as an image retrieval problem. This allows us to use readily available pretrained image similarity models which we can finetune.

We describe our finetuning datasets below in sec. V. We opted not to include several related datasets from the works

cited above: the From Anywhere to Object (FAO) [15] is more concerned with specific objects than locations, and the Where Are You? (WAY) [16] dataset uses dialogue rather than our choice of a single utterance. The relatively small size of these datasets (in the thousands of samples, on the order of our test set) suggests they may not be large enough to affect finetuning.

III. MATTERPORT3D ENVIRONMENT

Matterport3D is a collection of RGB-D images taken of indoor spaces by a Matterport panoramic camera. The dataset has 90 *scans* of buildings, mostly elaborate homes and a few oddities like cruise ship cabins and spas. Each *scan* is divided into a navigable graph of *viewpoints* or *views*, which are spread throughout at the house at a spacing of ~ 2.2 m. Finally, each *view* contains 36 RGB-D images, which can be knit together into a panoramic. We use equirectangular panoramics provided by Rey-Area *et al.* [18], although they were only able to reliably create images for about 85% of the *views* in the dataset. This limited our choice of data to those *views* we had coverage for (see Fig. 5).

IV. TASK AND ARCHITECTURE

We define our task in the Matterport3D environment. A user occupies one of M *views*, v_m , in a *scan*, s . The user gives a description, d , of their *view* to the agent. The agent has an image i_m for each $v \in s$, and produces a distribution

$$P(v_m | d, i_1, \dots, i_M) \quad \forall v \in s \quad (1 \leq m \leq M)$$

The agent may then guess which *view* the user occupies. A sample of our model’s output is shown by Fig. 3.

Defined this way, the problem is an *image-text similarity* task, suitable for large, pretrained transformer networks. OpenAI’s Contrastive Language-Image Pretraining (CLIP) [4], popular and readily available for finetuning, is a useful tool in vision and language tasks [19]. Recently, simple CLIP models have done very well in benchmarks like the RoboTHOR and Habitat ObjectNav challenges [20]. We used the `vit-base-patch32` model, the smaller vision transformer variant [21].

CLIP encodes N texts and M images with separately trained encoders. The dot product of the encodings are returned as (N, M) logit similarity scores (see Fig. 4).

To train CLIP, we encode a batch of N (text, image) pairs. A ‘perfect’ model produces a square (N, N) matrix which, after applying softmax, becomes the identity matrix. We use this matrix as a target for binary cross entropy loss and optimize the CLIP network with gradient descent. We used an Adam optimizer [22] with learning rate 5×10^{-7} and weight decay 10^{-3} . Finetuning took less than a day on an RTX 2080 Ti.

CLIP was pretrained on a large (10^8) dataset called WebImageText (WIT) [4]. On our human test set, described below, pretrained CLIP performs well above random as a zero-shot classifier. We use pretrained CLIP as a baseline measure in our experiments, and improve its performance through finetuning, described below.



Fig. 4. **Model.** CLIP [4] uses transformer networks [23], [24] to encode text and images into vectors of identical length, then compares these vectors by taking their dot product. In our task, a description might be compared with as many as 170 images (*views*) from the environment (*scan*).

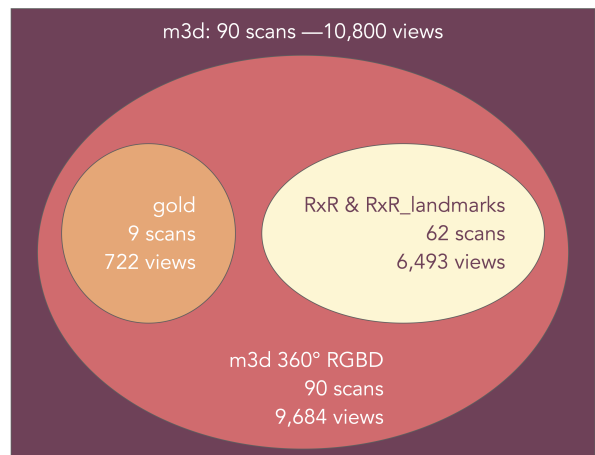


Fig. 5. (a) The Matterport3D (m3d) 360° RGBD set (red) contains most, but not all, of the images in Matterport3D (purple). We used the former for its equirectangular format. (b) The human ‘gold’ test set (orange) is disjoint from the finetuning sets RxR and RxR_landmarks (beige). Both are subsets of the m3d 360° RGBD set.

V. DATA

We used three datasets: a human test set which we collected, and the ‘RxR’ and ‘RxR_landmarks’ datasets, which we repurposed from existing data and used to finetune our model (see Table I and Fig. 5).

A. Human (‘gold’) set

To evaluate our model, we needed human descriptions of locations in Matterport3D. We chose 9 representative *scans* from the 90 total in Matterport3D, with 722 *views* between them. This covers about 10% of the Matterport3D environment. We collected two descriptions per *view*, for 1,443 samples total. Different humans wrote each description.

We used Amazon Mechanical Turk (AMT) to collect our data (see Fig. 7). For each sample, we show the worker a skybox image of a *view* and ask them to describe their

TABLE I
DATA STATISTICS ('M3D' REFERS TO MATTERPORT3D)

set	# scans	# views	# samples	avg # words
gold	9	722	1,443	23
RxR	62	6,493	205,092	14
RxR_landmarks	62	6,472	172,309	2
m3d 360° RGBD	90	9,684	n/a	n/a
m3d	90	10,800	n/a	n/a



RxR	RxR_landmarks
"you can see a big brown colour table on your right side and two brown colour chairs and white pillars on your left"	"first white pillar"
"have reached you're destination point"	"side walk"
"left and"	"two bathtubs"

Fig. 6. We drew from two existing datasets to create a finetuning set for CLIP. RxR, an instruction dataset, generally provided longer examples with more extensive grammar, but many fragments were unrelated to the image. RxR_landmarks, on the other hand, provided very relevant, but very short, keyword samples.

location, so that another person who knew the space could find them.

B. Room Across Room ('RxR')

RxR [6] is a navigation dataset in the Matterport3D environment. It contains, among other things, human descriptions of paths in *scans*. Each path is a sequence of *views*. Human 'guides' were asked to describe their journey along this path so that another user could follow it.


RxR has been used to evaluate language grounding models [6], and most recently to generate instructions for human evaluation [7], but not for our particular task of location. However, it is a high quality source of captioned imagery specific to our environment, so we used it to finetune our model.

Samples in RxR contain a series of panoramic images (the path) and guide annotations describing the path. Each word in the annotation is time stamped, allowing us to map it to a *view* along the path, giving us 205,092 (description, *view*) pairs (see Fig. 6).

RxR includes image masks to indicate what parts of a *view* were visible to the guide when they said the word, but we omitted those masks here.

Instructions:

Please imagine that you're in a house. This is what you can see.



Please describe where you are.

Please use enough detail that someone else in the house could figure out exactly where you are. For example, you might mention specific objects or colors.

Try to give us two or three sentences if you can.

Grammar isn't super important, but please be careful of spaces and spelling.

a) collection of gold dataset



"There are two lights on the ceiling. In front of the gray padded seat are two small rectangular tables."

"The room has silver cushioned walls and large lounge chairs."

b) gold data examples

We are asking you to pick a location in a house based on a text description.


Some of the hints can be pretty hard, and some of the images look alike, so it's okay if you're not perfect.

Please don't guess randomly, because we can't approve random answers.

Instructions

Please choose the image that most closely matches this description:

I'm behind several white chairs. On my left side are windows with blinds and transparent white curtains.



c) human evaluation of gold dataset

Fig. 7. We collected an evaluation dataset by asking users of Amazon Mechanical Turk (AMT) to describe their surroundings when shown panoramas from the Matterport3D dataset. (a) Our 'gold' dataset has two samples for each *view* in nine buildings, about 10% of the total Matterport3D set. (b) We evaluated each sample by asking another AMT worker to pick the correct image from a collection of 20. (c) Each 'gold' sample was evaluated once.

C. Room Across Room 'landmarks'

Wang *et al.* [7] developed the 'landmarks' dataset from the original RxR data to help generate instructions to guide users in the Matterport3D environment. It contains 172,309 samples. For each guide annotation of a path in RxR, Wang *et al.* first used a transformer to identify 'entities' in the text, for example a couch, bathtub, or door. They associated these entities with specific regions (bounding boxes) of panoramic

TABLE II
COMPARING MODELS ACROSS ENTIRE GOLD DATASET

model	data	success (%) \uparrow	hits at 1 (%) \uparrow	close (%) \uparrow	same room (%) \uparrow	error (m) \downarrow	mrr \uparrow
random		11	1	7	8	15.73	0.06
CNN-LSTM	rxr + landmarks	11	1	7	9	16.68	0.06
CLIP	no pretrain, rxr + landmarks	14	3	11	11	15.75	0.09
CLIP	pretrain only	44	10	34	36	10.10	0.23
CLIP	pretrain + landmarks	47	12	37	41	8.75	0.26
CLIP	pretrain + rxr	55	14	43	46	7.74	0.28
CLIP	pretrain + rxr + landmarks	55	14	44	47	7.30	0.28

images along the path.

The resulting (landmark, region) pairs were helpful in generating instructions to guide human users. We use this data directly as our RxR_landmarks dataset (see Fig. 6). We omit the bounding boxes, so that our data simply maps entities to entire *views*.

VI. EVALUATION AND RESULTS

In this section, we describe our metrics for evaluating our model, and give the results of two experiments: a comparison of different finetuning datasets and models on the full task, and then a comparison of our best model against a human baseline on a smaller, easier version of our task.

A. Metrics

- 1) **success (%)**: Percent success rate. The model is ‘successful’ if its guess satisfies one of the following:
 - *hits at 1 (%)*: The guess is the target image.
 - *close (%)*: The guess is less than 3m from the target image, in graph distance. The average spacing in Matterport3D is ~ 2.2 m, so 3m is a common success threshold in related literature using this environment. All *hits at 1* pass this test.
 - *same room (%)*: Matterport3D *scans* are annotated by hand into ‘regions’ like bathroom, living room, etc. We say that the model is successful if it was able to guess the correct region. All *hits at 1* pass this test.

Fig. 3 shows some of these success conditions.

- 2) **error (m)**: The average graph distance between the guess and the target *view*.
- 3) **mean reciprocal rank (MRR)**: $\frac{1}{k} \sum_k \frac{1}{\text{rank}}$, where *rank* is the priority of the target in the model’s confidence distribution. An MRR of 0.5 indicates that the model is expected to rank the target image second.

B. Comparing Models and Finetuning

In our first experiment, we compared different models on the original (all *views*) task (see Table II). The random baseline chooses one of M *views* in the *scan*. The CNN-LSTM model encodes the *view* image with a pretrained ResNet152 [25] and the description with a bidirectional, 3-layer LSTM encoder using the CLIP tokenizer.

The ‘no pretrain’ model is a CLIP that is trained only on the combined finetuning data (no WIT), whereas ‘pretrain only’ is CLIP only trained on its WIT dataset. Finally, we show some ablations of the finetuning dataset, comparing

the relative contribution of RxR and RxR_landmarks to the combined dataset in the final row.

We show an example of our model’s output in Fig. 3. As the caption describes, the model is frequently able to identify key features from the description. But even when it does identify the right *type* of room, it might fail to pick the correct instance of that type – a particular bathroom or bedroom, for example.

This experiment shows the importance of CLIP’s pretraining. The model without pretraining (Table II row 3) performs little better than random (row 1). Our finetuning dataset, at 10^5 samples, is probably too small to train a large model like CLIP. Similarly, the CNN-LSTM variant (row 2) performs only at random, despite the pretrained ResNet.

Finetuning CLIP (rows 5–7) increased the pretrained model’s performance (row 4) by up to 25% on the test set. This is a substantial gain, given that the finetuning set is three orders of magnitude smaller than the pretraining data.

We were surprised to see that the RxR_landmarks set seems to add little beyond the RxR set. The one or two word descriptions of landmarks are much shorter than our test set queries, which are several sentences long. Most of its landmark information is probably present in the RxR set.

C. Human Baseline

To test our model against a human baseline, we scaled down our task. In our task defined above, the model compares the description to every *view* in the *scan*. A *scan* might have well over a hundred *views*. Our model can process an arbitrary number of *views*, but comparing that many similar images is very challenging for a human evaluator to do.

Instead, we presented workers on AMT with only 20 *views*, randomly chosen from the same *scan*, one of which is the true target *view* which the query text describes.¹ We collected one evaluation for each sample in the *gold* dataset (1,443), and scored them according to the metrics above. To control the quality of responses, we discarded data from workers who failed to score a ‘success’ in at least 20% of their responses, which is above random (12%).

We compared our best finetuned CLIP model (Table III row 4) to the human baseline (row 3). Like the humans, the model chose from only 20 images. In this experiment, the model slightly outperformed the human baseline. However,

¹The Northeastern University IRB has reviewed this research and exempted it from further action.

TABLE III
COMPARISON TO HUMAN BASELINE (ONLY 20 CHOICES)

model	data	success (%) \uparrow	hits at 1 (%) \uparrow	close (%) \uparrow	same room (%) \uparrow	error (m) \downarrow
random		12	4	10	10	15.70
CLIP	pretrain only	51	30	45	45	8.10
human		57	38	52	53	7.07
CLIP	pretrain + rxr + landmarks	63	38	54	58	5.90

we address some limitations of the human evaluation, and other aspects of our work, in the next section.

VII. LIMITATIONS, AND FUTURE DIRECTIONS

A. Human Evaluation

We used AMT to get a human baseline on our task. However, performance between workers varied significantly: Some individuals scored a success rate of 60–70%, others between 20–30%. This disparity was expected, given that humans vary in their ability to navigate and track spatial relationships. [26]

An informal pilot study showed that graduate students scored about a 70% success rate on the same task. These students were very familiar with the dataset and the task, so their performance might be a better estimate of good human performance. However, we lacked the resources to evaluate the entire test set this way.

Both the informal and formal figures (70% and 57%) may seem low for a human evaluation. We suggest two explanations. First, humans were asked to choose from 20 independent images; this is a somewhat unnatural task and not how humans generally think of spatial environments. Second, some descriptions of the environment are ambiguous and could easily refer to more than one location. The example in fig. 3 shows that houses tend to have several bathrooms which all look quite similar. Given these challenges, we think the human success rate is reasonable, and compares well to human performance on a similar location task in the same environment (70.4% for human locators in WAY [16]).

B. Continuous Environment

Human performance might increase with a better representation of our environment. Our Matterport3D environment uses a navigable graph of *views*, but more recent implementations use a continuous environment compatible with a simulator like Habitat [10], [27].

In a continuous environment, the user can move more naturally throughout a building, exploring and stopping the simulation when they think they have found the right location. This formulation of the task is more natural. In addition to being easier for humans to understand, a continuous training environment would help a robot prepare better for a real world demonstration.

C. Improving Data

A simpler improvement would be to use bounding box and mask information for our finetuning datasets. As mentioned earlier, the RxR datasets (including RxR_landmarks)

have image mask and bounding box data that might allow us to draw a tighter relationship between text and image regions. We omitted these bounding boxes, so that in the RxR_landmarks set for example, the word ‘refrigerator’ is mapped to an entire panoramic image of a kitchen. The model may perform better with a more specific mapping of concepts to pixels.

In addition to the bounding box information from the datasets we used, we expect that finetuned results would improve with additional relevant data, including those we described in section II.

VIII. CONCLUSION

We proposed the task of vision-language localization, and considered a simple approach using existing pretrained vision-language models. We also collected and constructed several datasets for this task, to finetune the pretrained model. We outperformed the baseline pretrained model, as well as a human baseline on a simplified version of our task. In future work, we plan to deploy this competitive approach on a robot operating in a continuous environment, to locate users using natural language.

IX. ACKNOWLEDGEMENTS

This work was supported by NSF Grants #2107256 and #2142519.

REFERENCES

- [1] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, “Robots that use language,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 25–55, 2020.
- [2] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3D: Learning from RGB-D data in indoor environments,” *International Conference on 3D Vision (3DV)*, 2017.
- [3] Y. Du, Z. Liu, J. Li, and W. X. Zhao, “A survey of vision-language pre-trained models,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.
- [4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning (ICML)*, 2021.
- [5] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, “Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [6] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, “Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding,” in *Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2020.

- [7] S. Wang, C. Montgomery, J. Orbay, V. Birodkar, A. Faust, I. Gur, N. Jaques, A. Waters, J. Baldrige, and P. Anderson, "Less is more: Generating grounded navigation instructions from landmarks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [8] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. Wang, "Vision-and-language navigation: A survey of tasks, methods, and future directions," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.
- [9] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, "VLN²BERT: A recurrent vision-and-language BERT for navigation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [10] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *European Conference on Computer Vision (ECCV)*, 2020.
- [11] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, "History aware multimodal transformer for vision-and-language navigation," *Neural Information Processing Systems (NeurIPS)*, 2021.
- [12] J. Krantz, S. Banerjee, W. Zhu, J. Corso, P. Anderson, S. Lee, and J. Thomason, "Iterative vision-and-language navigation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [13] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [14] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. v. d. Hengel, "REVERIE: Remote embodied visual referring expression in real indoor environments," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [15] F. Zhu, X. Liang, Y. Zhu, Q. Yu, X. Chang, and X. Liang, "SOON: Scenario oriented object navigation with graph-based exploration," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [16] M. Hahn, J. Krantz, D. Batra, D. Parikh, J. Rehg, S. Lee, and P. Anderson, "Where are you? localization from embodied dialog," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.
- [17] D. Z. Chen, A. X. Chang, and M. Nießner, "ScanRefer: 3D object localization in RGB-D scans using natural language," in *European Conference on Computer Vision (ECCV)*, 2020.
- [18] M. Rey-Area, M. Yuan, and C. Richardt, "Matterport3D 360° RGBD dataset," March 2022. [Online]. Available: <https://researchdata.bath.ac.uk/1126/>
- [19] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao, and K. Keutzer, "How much can CLIP benefit vision-and-language tasks?" in *International Conference on Learning Representations (ICLR)*, 2022.
- [20] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: CLIP embeddings for embodied AI," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [21] "openai-clip-vit-base-patch32." [Online]. Available: <https://huggingface.co/openai/clip-vit-base-patch32>
- [22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems (NeurIPS)*, 2017.
- [24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [26] M. Kozhevnikov, M. A. Motes, B. Rasch, and O. Blajenkova, "Perspective-taking vs. mental rotation transformations and how they predict spatial navigation performance," *Applied Cognitive Psychology*, vol. 20, no. 3, pp. 397–417, 2006.
- [27] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat:

A platform for embodied AI research," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.